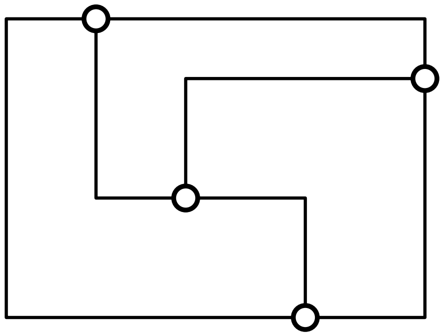
Sketched Representations and Orthogonal Planarity of Bounded Treewidth Graphs

Emilio Di Giacomo, Giuseppe Liotta, Fabrizio Montecchiani
University of Perugia, Italy

GD 2019, September 17-20, 2019, Průhonice/Prague

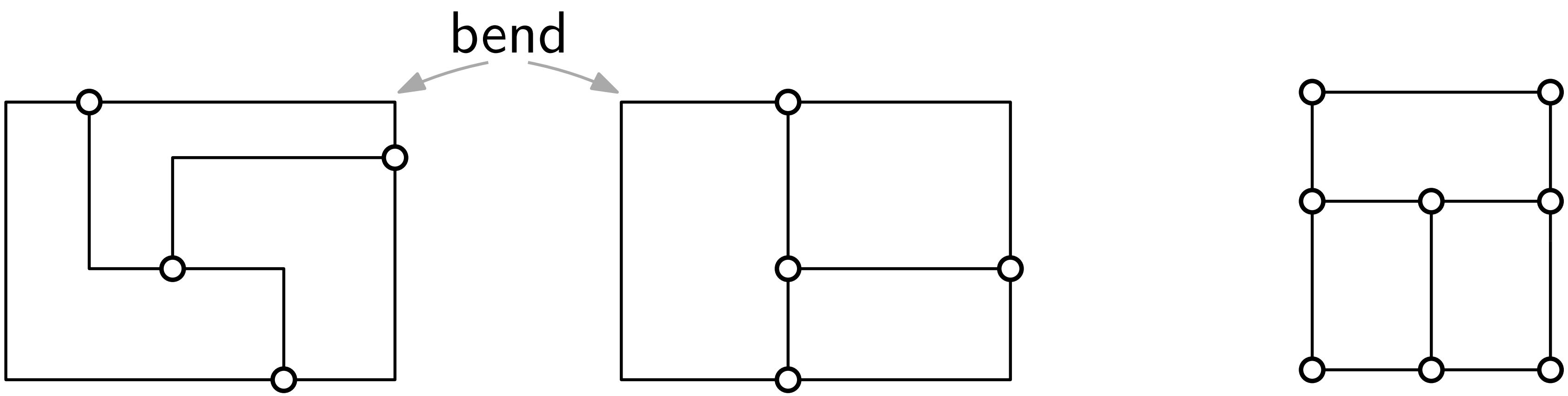
Orthogonal Drawings

An orthogonal drawing of a planar graph with max degree 4 is a planar drawing where each edge is drawn as a chain of horizontal and vertical segments.



Orthogonal Drawings

An orthogonal drawing of a planar graph with max degree 4 is a planar drawing where each edge is drawn as a chain of horizontal and vertical segments.



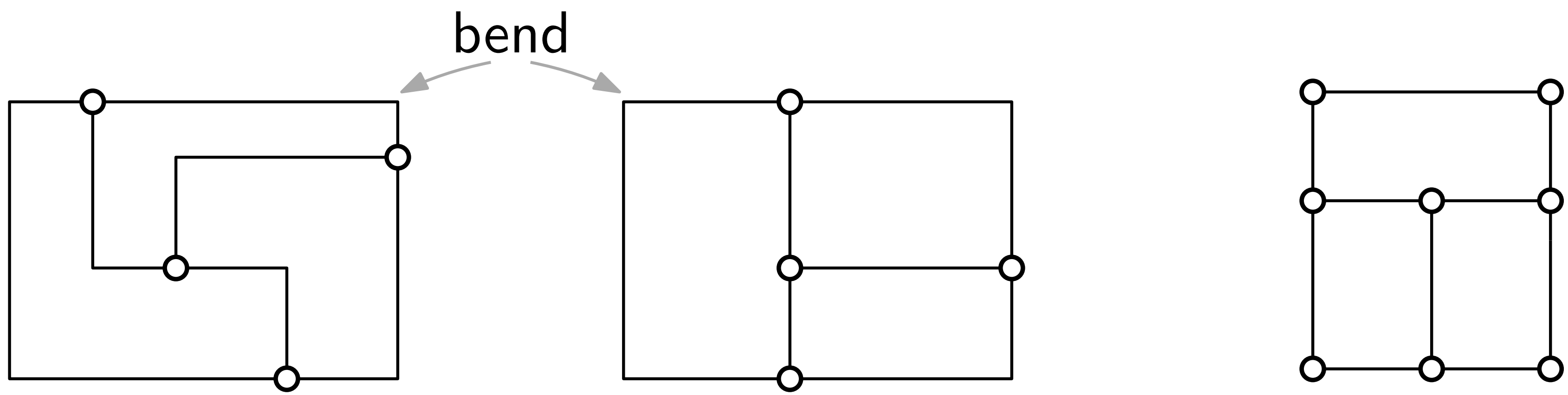
Orthogonal drawing with 7 bends

Orthogonal drawing with 4 bends

Rectilinear (no bends) drawing

Orthogonal Drawings

An orthogonal drawing of a planar graph with max degree 4 is a planar drawing where each edge is drawn as a chain of horizontal and vertical segments.



Orthogonal drawing with 7 bends

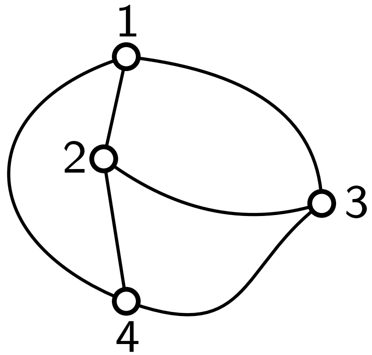
Orthogonal drawing with 4 bends

Rectilinear (no bends) drawing

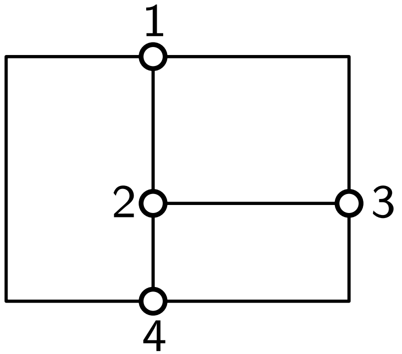
Every planar graph with max degree 4 (except the octahedron) admits an orthogonal drawing with at most 2 bends per edge and at most $\mathbf{2n + 2}$ **bends in total** [Biedl & Kant 1998].

Orthogonal Planarity

ORTHOGONALPLANARITY: Given a planar graph G and an integer b, does G admit an orthogonal drawing with at most b bends?



$\langle G, b = 4 \rangle$

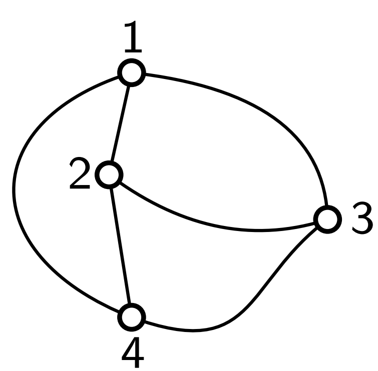


Γ

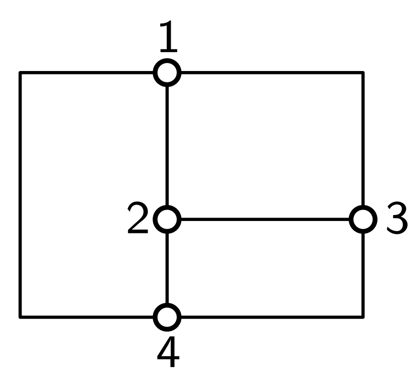
Orthogonal Planarity

OrthogonalPlanarity: Given a planar graph G and an integer b, does G admit an orthogonal drawing with at most b bends?

- **NP-complete** for $b = 0$ [Garg & Tamassia 2001].
  - **NP-hard** to approximate the minimum number of bends with an $O(n^{1-\varepsilon})$ error ($\varepsilon > 0$) [Garg & Tamassia 2001].
- If G is 2-connected, FPT algorithm parametrized by the number of degree-4 vertices [Didimo & Liotta 1998].
- $O(n^4)$-time algorithm for series-parallel graphs [Di Battista, Liotta, Vargiu 1998].



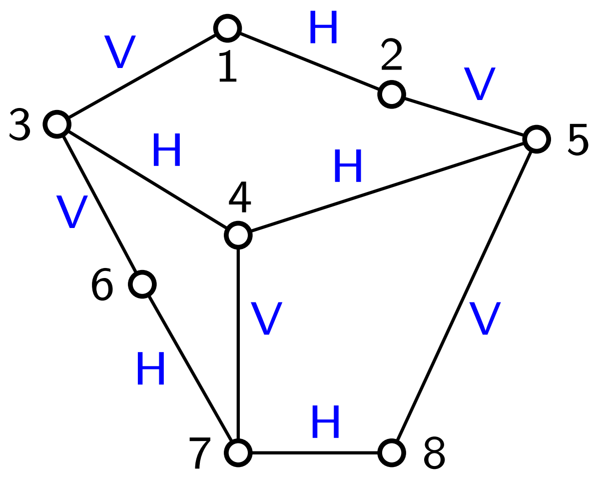
$\langle G, b = 4 \rangle$

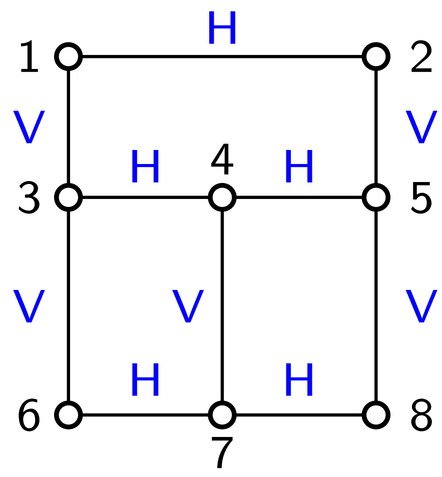


Γ

Related Problems

HV-PLANARITY: Given planar graph G whose edges are each labeled H (horizontal) or V (vertical), does G admit a rectilinear drawing in which edge directions are consistent with their labels?

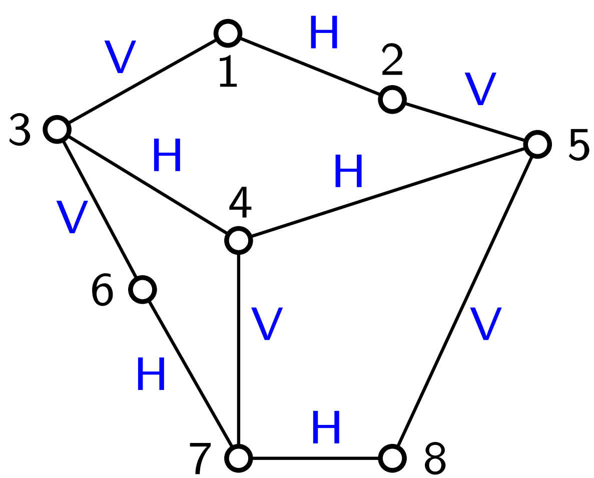


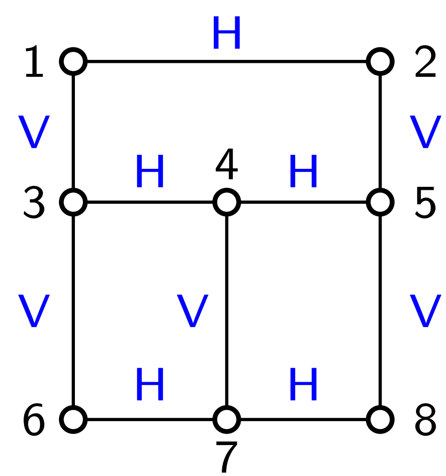


Related Problems

HV-PLANARITY: Given planar graph G whose edges are each labeled H (horizontal) or V (vertical), does G admit a rectilinear drawing in which edge directions are consistent with their labels?

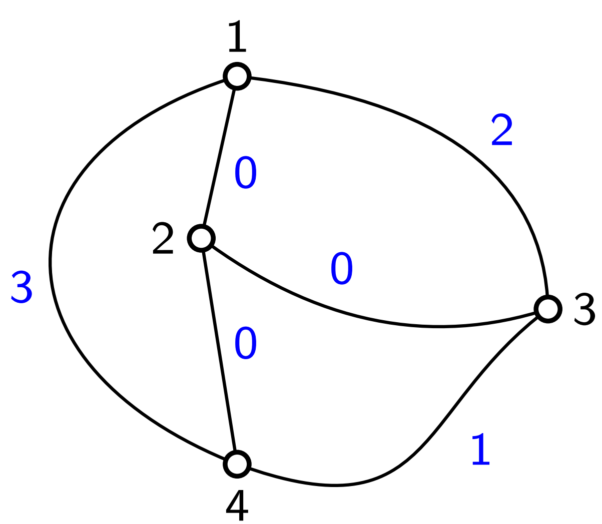
- **NP-complete** [Didimo, Liotta, Patrignani 2019].
- $O(n^4)$-time algorithm for series-parallel graphs [Didimo, Liotta, Patrignani 2019].

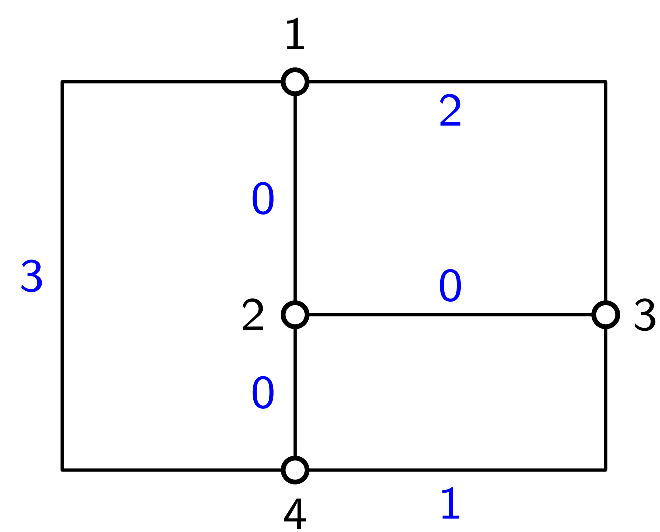




Related Problems

FLEXDRAW: Given a planar graph G whose edges have integer weights, does G admit an orthogonal drawing where each edge has a number of bends that is at most its weight?

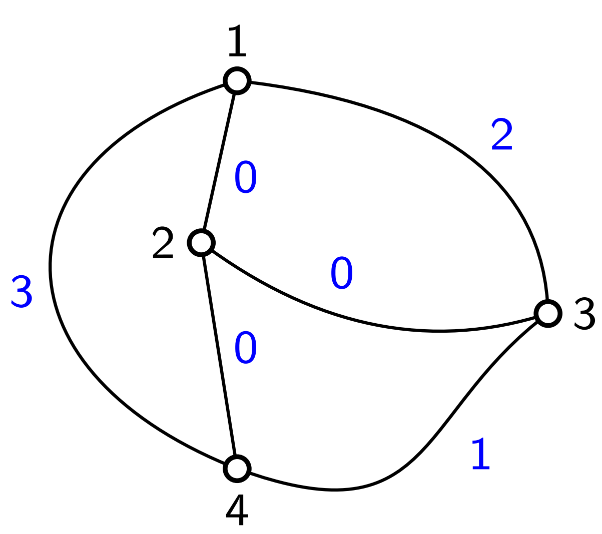


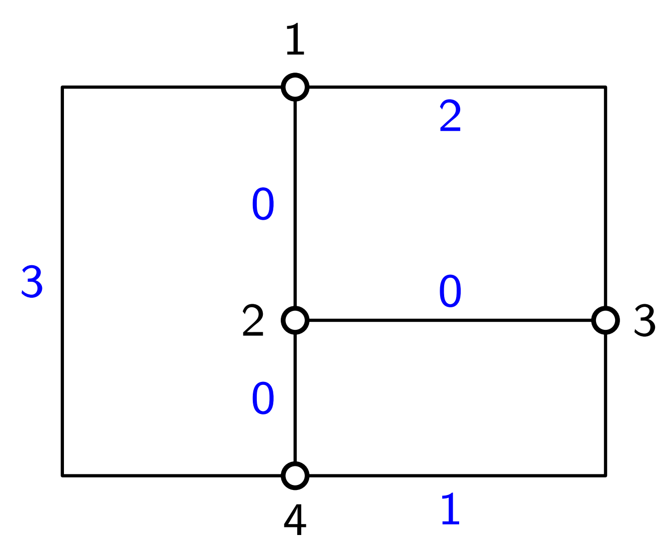


Related Problems

FLEXDRAW: Given a planar graph G whose edges have integer weights, does G admit an orthogonal drawing where each edge has a number of bends that is at most its weight?

- **NP-complete** [Garg & Tamassia 2001; Bläsius, Krug, Rutter 2014].
- $O(n^2)$-time algorithm if weights are positive [Bläsius, Krug, Rutter 2014].
- FPT algorithm parametrized by the number of edges that cannot be bent [Bläsius, Lehmann, Rutter 2016].





Contribution

Contribution

Main Theorem: ORTHOGONALPLANARITY (HV-PLANARITY, FLEXDRAW) can be solved in polynomial time for graphs of bounded treewidth.

- The problems lie in the XP class parameterized by treewidth (time complexity is $n^{g(k)}$, where k is the treewidth).
- Can be used to minimize bends.

Contribution

Main Theorem: OrthogonalPlanarity (HV-Planarity, FlexDraw) can be solved in polynomial time for graphs of bounded treewidth.

- The problems lie in the XP class parameterized by treewidth (time complexity is $n^{g(k)}$, where k is the treewidth).
- Can be used to minimize bends.

Corollary: OrthogonalPlanarity (HV-Planarity) can be solved in $O(n^3 \log n)$ time for series-parallel graphs.

- Improves on previous $O(n^4)$ bounds [Di Battista, Liotta, Vargiu 1998; Didimo, Liotta, Patrignani 2019].

Proof Ideas & Tools

An FPT Algorithm

- Constructive proof based on FPT algorithm with parameters: treewidth k, num. of degree-2 vertices σ and num. of bends b.

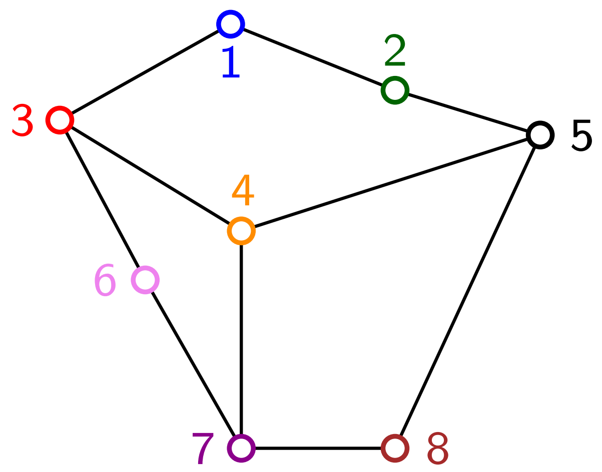
An FPT Algorithm

- Constructive proof based on FPT algorithm with parameters: treewidth k, num. of degree-2 vertices σ and num. of bends b.

Fine-grained Theorem: Let G be an n-vertex planar graph. Given a tree-decomposition of G of width k, there is an algorithm that decides $\textsc{OrthogonalPlanarity}$ in $f(k, \sigma, b) \cdot n$ time, where $f(k, \sigma, b) = k^{O(k)}(\sigma + b)^k \log(\sigma + b)$.
The algorithm computes a drawing of G, if one exists.

Tree-decomposition

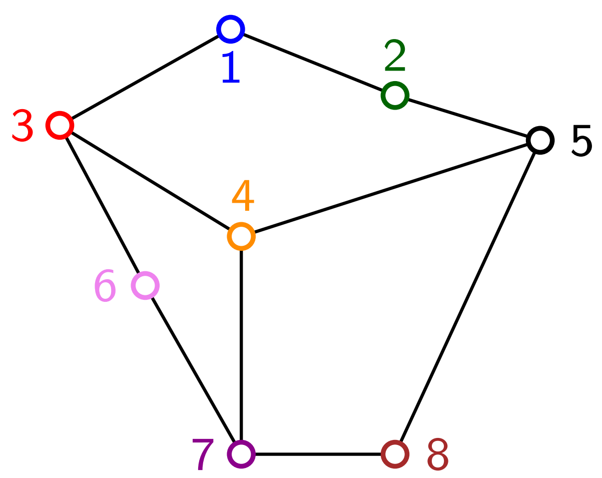
A tree-decomposition of a graph $G = (V, E)$ is a pair $(\mathcal{X}, T)$ s.t.:

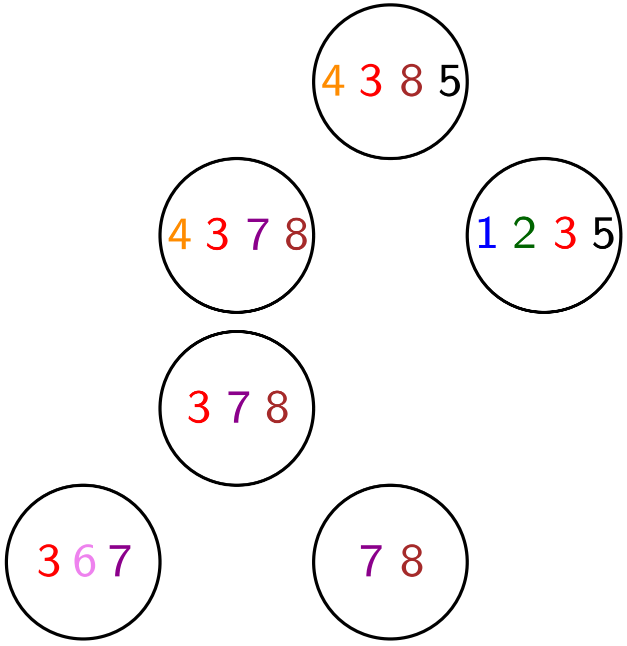


Tree-decomposition

A tree-decomposition of a graph $G = (V, E)$ is a pair $(\mathcal{X}, T)$ s.t.:

- $\mathcal{X} = \{X_1, X_2, \ldots, X_\ell\}$ is a set of subsets of V called bags,

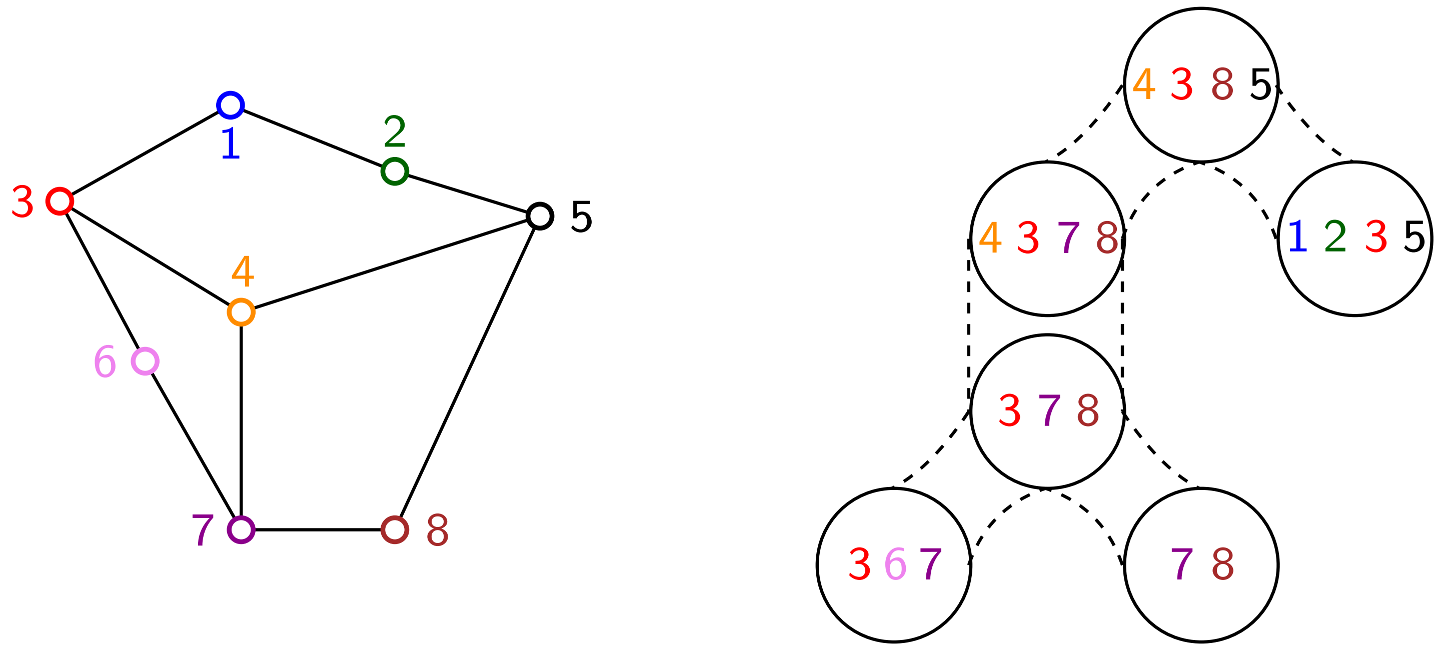




Tree-decomposition

A tree-decomposition of a graph $G = (V, E)$ is a pair $(\mathcal{X}, T)$ s.t.:

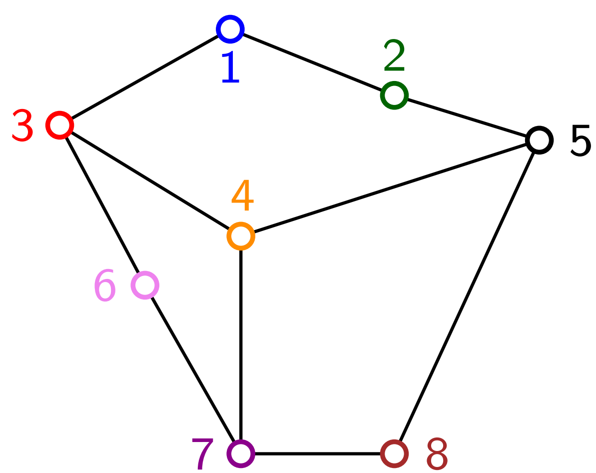
- $\mathcal{X} = \{X_1, X_2, \ldots, X_\ell\}$ is a set of subsets of V called bags,
- T is a tree, each node is mapped to a bag of $\mathcal{X}$,

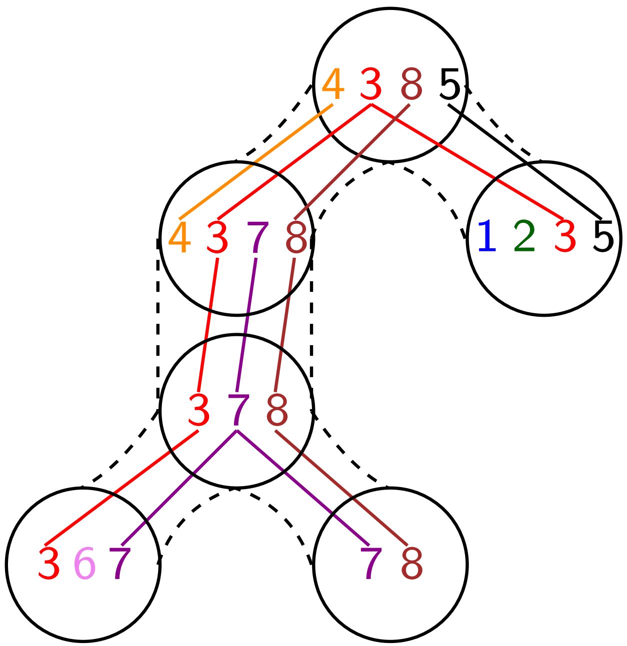


Tree-decomposition

A tree-decomposition of a graph $G = (V, E)$ is a pair $(\mathcal{X}, T)$ s.t.:

- $\mathcal{X} = \{X_1, X_2, \ldots, X_\ell\}$ is a set of subsets of V called bags,
- T is a tree, each node is mapped to a bag of $\mathcal{X}$,
- $\forall (u, v) \in E$, $\exists X_i : u, v \in X_i$,
- $\forall v \in V$ the bags containing v induces a non-empty subtree of T



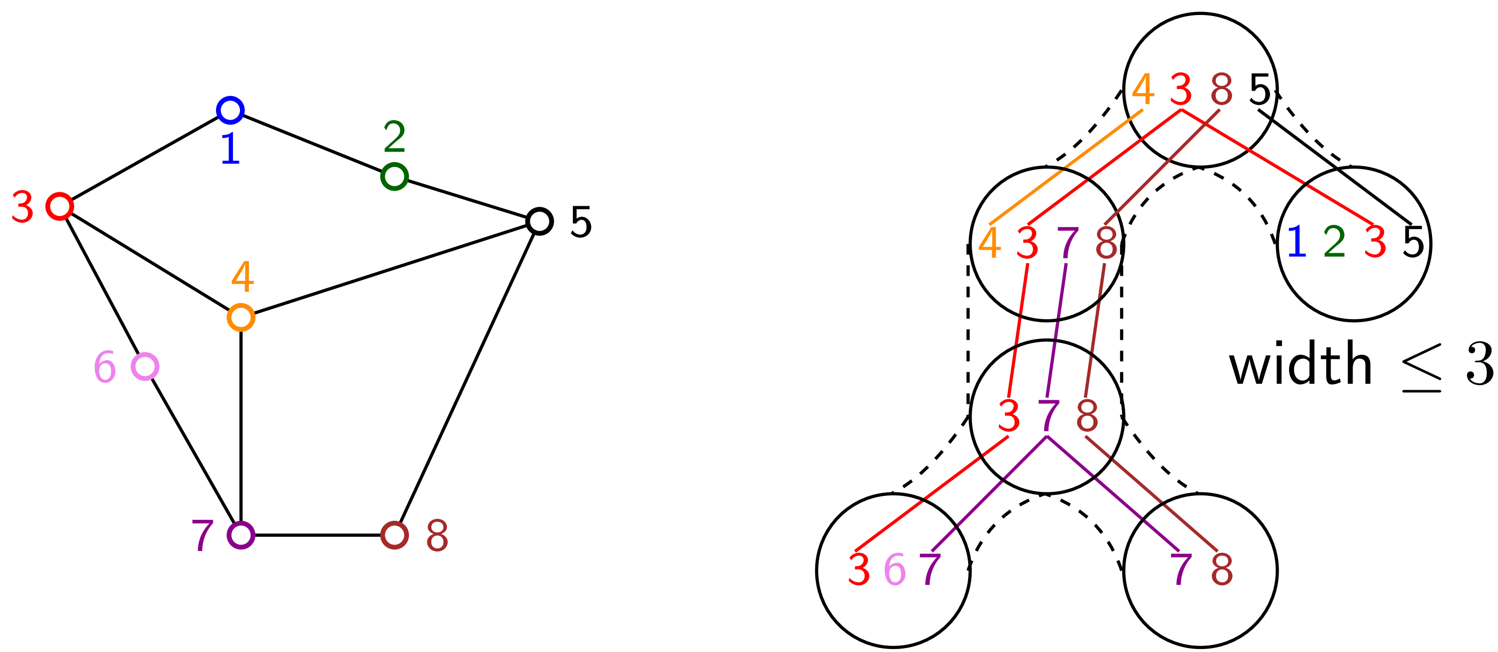


Tree-decomposition

A tree-decomposition of a graph $G = (V, E)$ is a pair $(\mathcal{X}, T)$ s.t.:

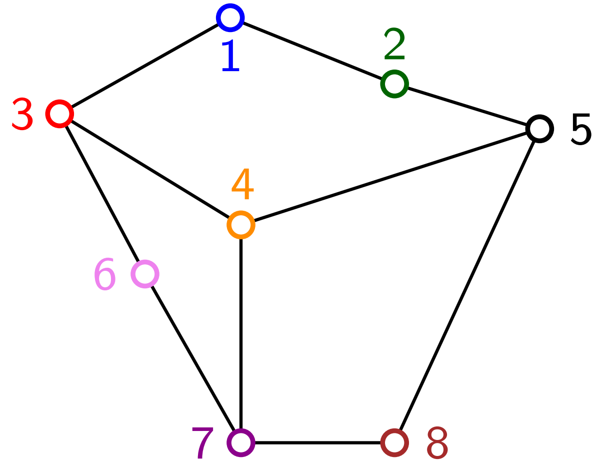
- $\mathcal{X} = \{X_1, X_2, \ldots, X_\ell\}$ is a set of subsets of V called bags,
- T is a tree, each node is mapped to a bag of $\mathcal{X}$,
- $\forall (u, v) \in E$, $\exists X_i : u, v \in X_i$,
- $\forall v \in V$ the bags containing v induces a non-empty subtree of T

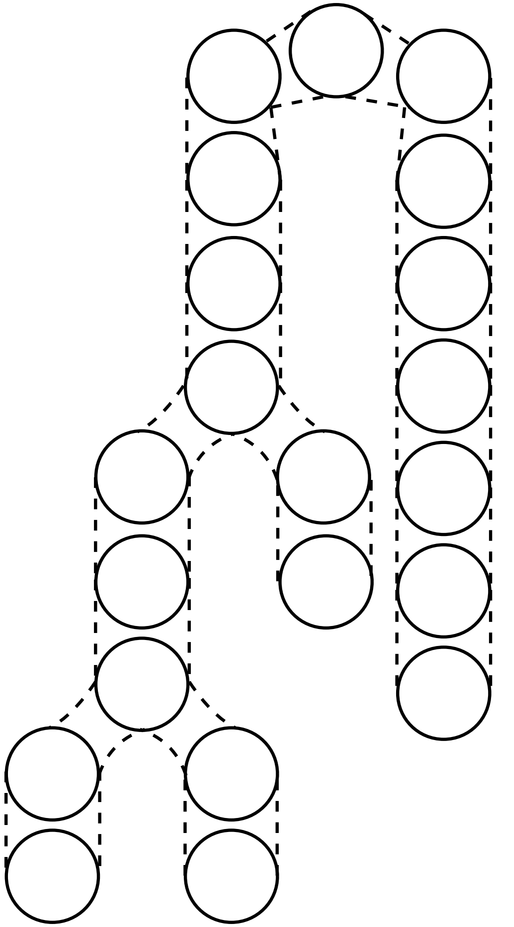
The width of $(\mathcal{X}, T)$ is $\max_{i=1}^{\ell} |X_i| - 1$.
The treewidth of G is the minimum width over all its tree-decompositions.



Nice Tree-decomposition

$(\mathcal{X}, T)$ is nice if T is a binary rooted tree and:

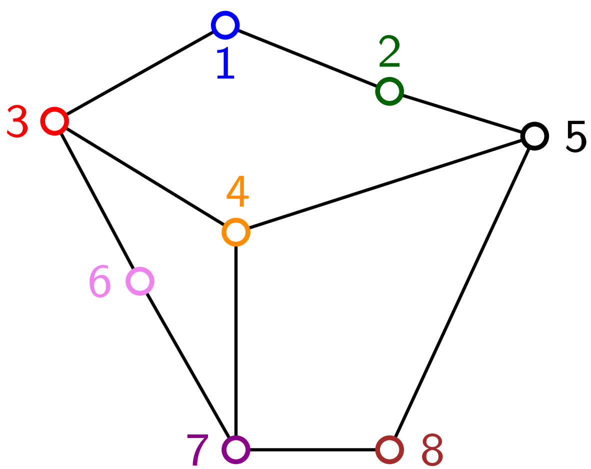


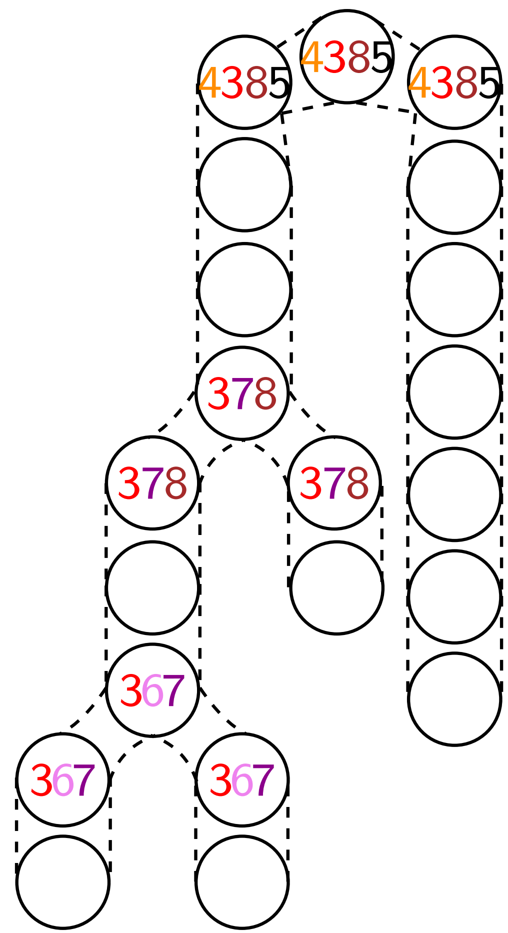


Nice Tree-decomposition

$(\mathcal{X}, T)$ is nice if T is a binary rooted tree and:

- **(JOIN)** if X_i has 2 children X_j and $X_{j'}$ then $X_i = X_j = X_{j'}$

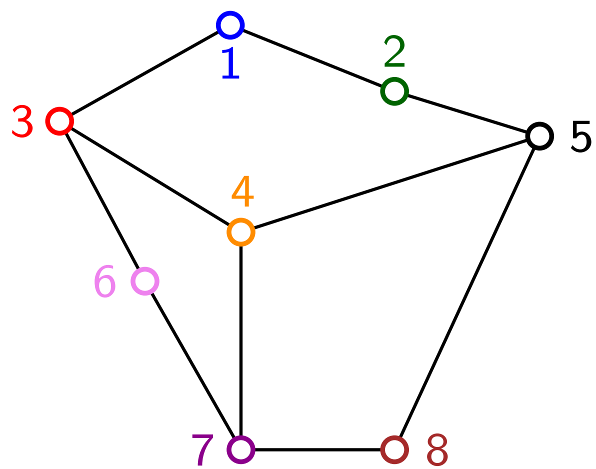


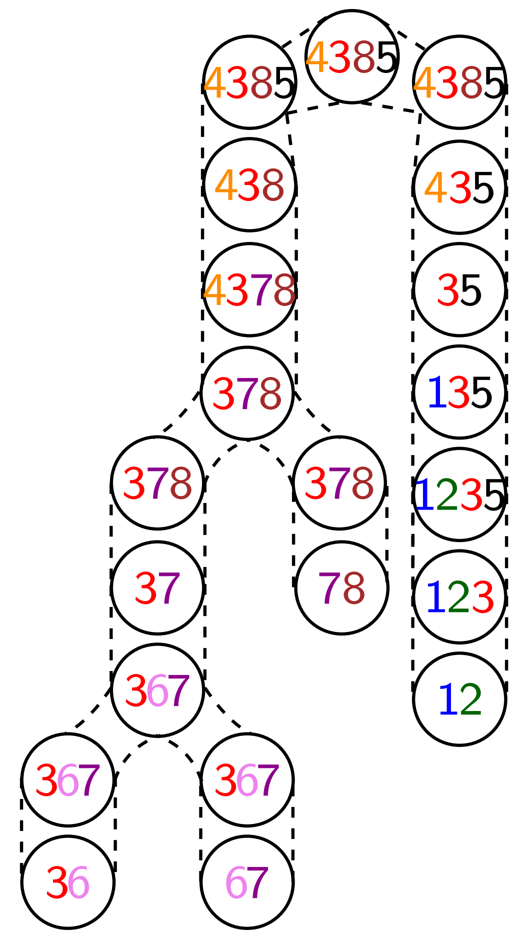


Nice Tree-decomposition

$(\mathcal{X}, T)$ is nice if T is a binary rooted tree and:

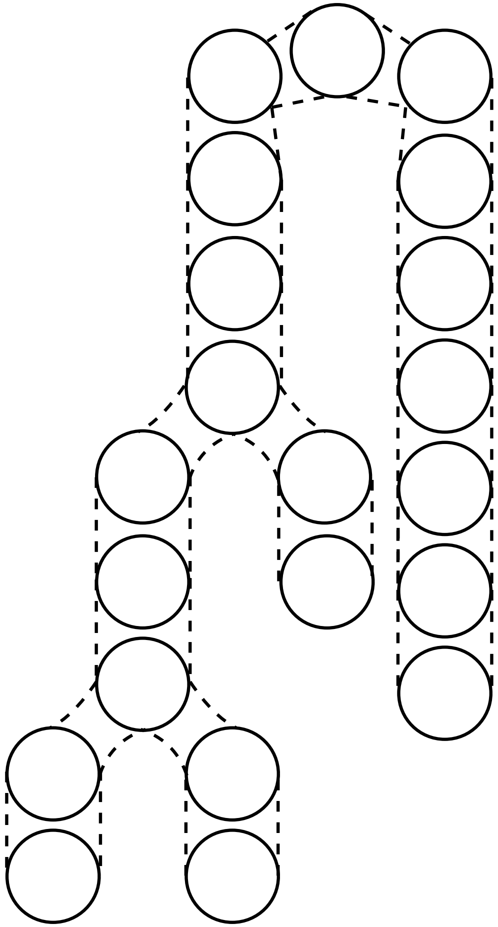
- (**JOIN**) if X_i has 2 children X_j and $X_{j'}$ then $X_i = X_j = X_{j'}$
- if X_i of T has only 1 child X_j then there is $v \in V$ s.t. either
  - (**INTRODUCE**) $X_i = X_j \cup \{v\}$, or
  - (**FORGET**) $X_i \cup \{v\} = X_j$





Nice tree decomposition

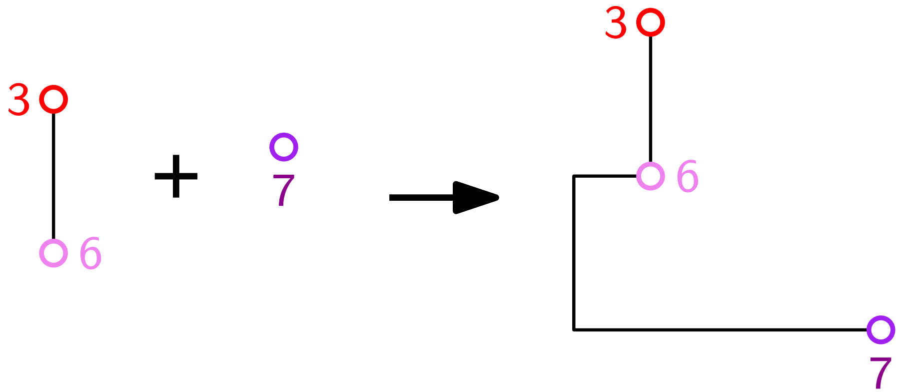
IDEA: design a dynamic programming algorithm based on a nice tree-decomposition of the graph

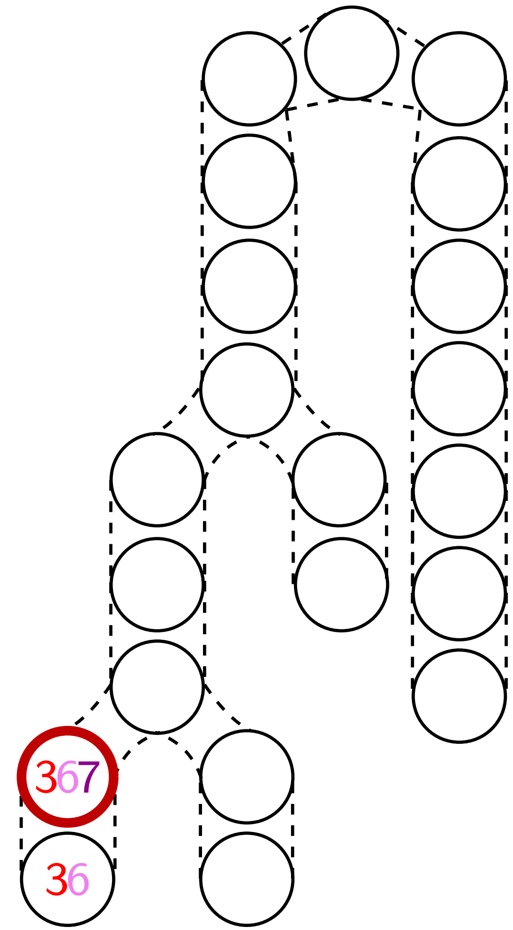


Nice tree decomposition

IDEA: design a dynamic programming algorithm based on a nice tree-decomposition of the graph

INTRODUCE vertex in all drawings

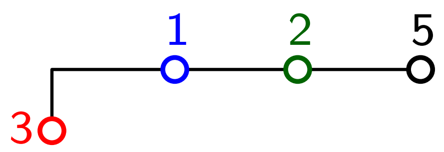


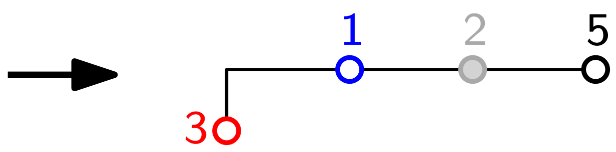


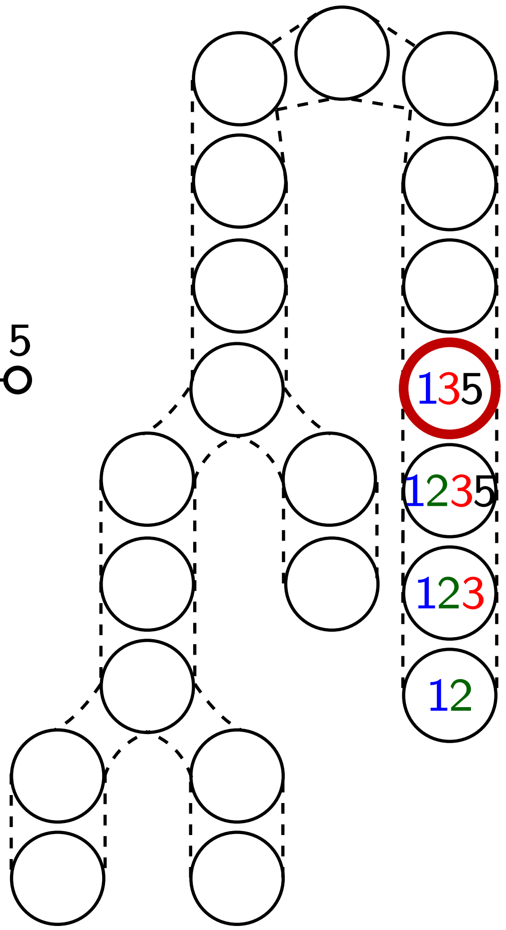
Nice tree decomposition

IDEA: design a dynamic programming algorithm based on a nice tree-decomposition of the graph

FORGET (DEACTIVATE) vertex in all drawings



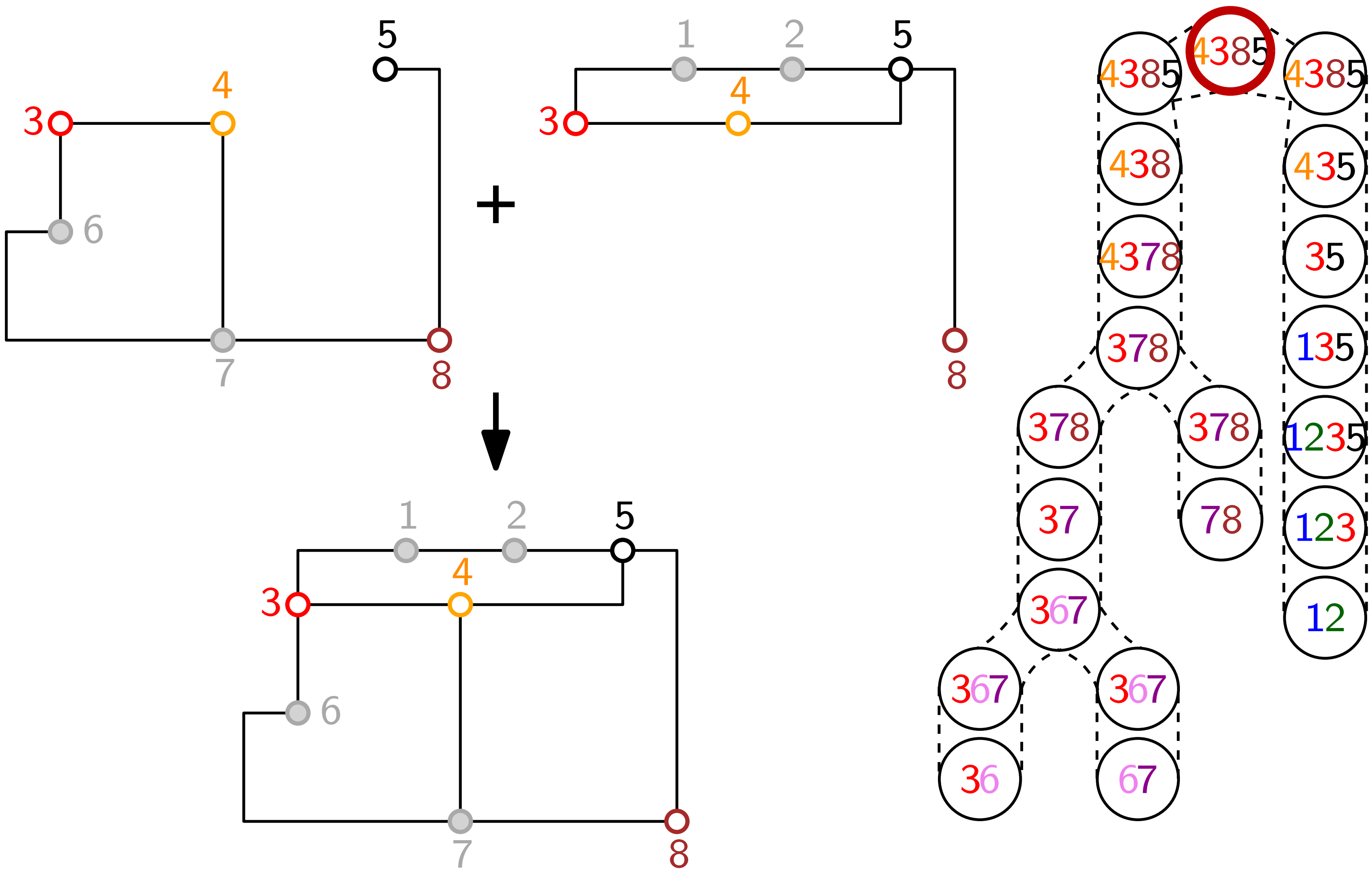




Nice tree decomposition

IDEA: design a dynamic programming algorithm based on a nice tree-decomposition of the graph

JOIN all pairs of drawings



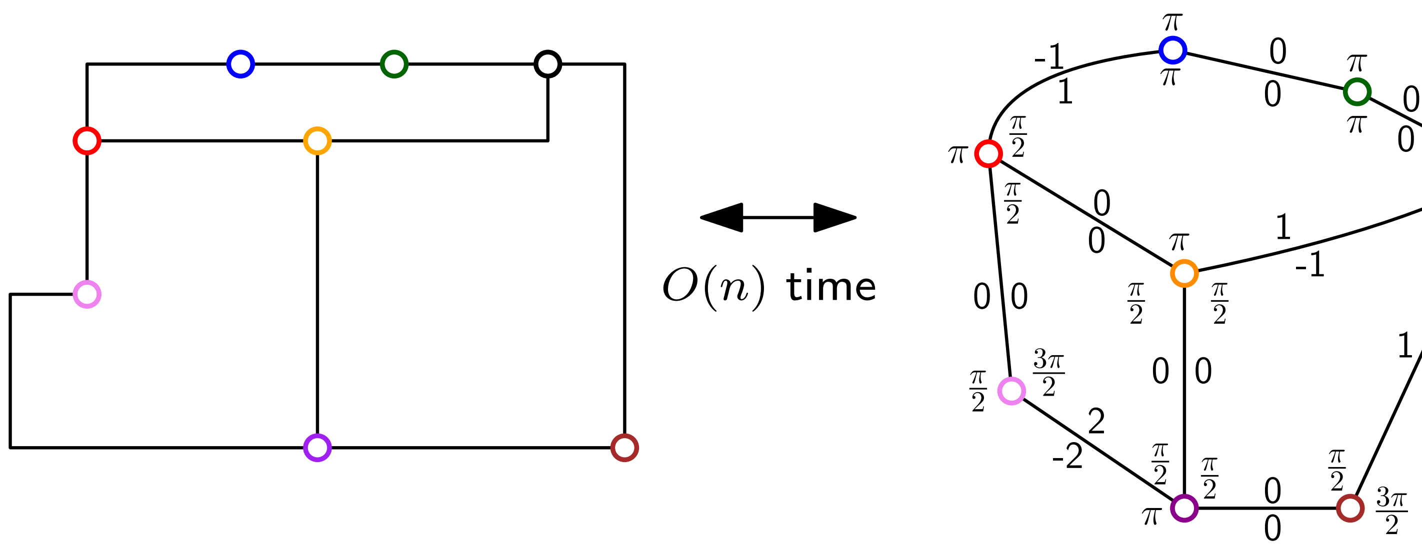
Nice tree decomposition

IDEA: design a dynamic programming algorithm based on a nice tree-decomposition of the graph

TODO: design abstract (small) records to replace drawings!

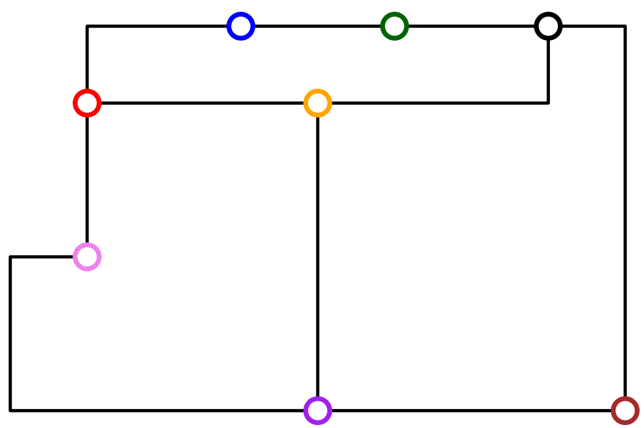
Orthogonal Representations

An orthogonal representation of a plane graph G represents an equivalence class of orthogonal drawings with the same “shape” [Tamassia, 1987].

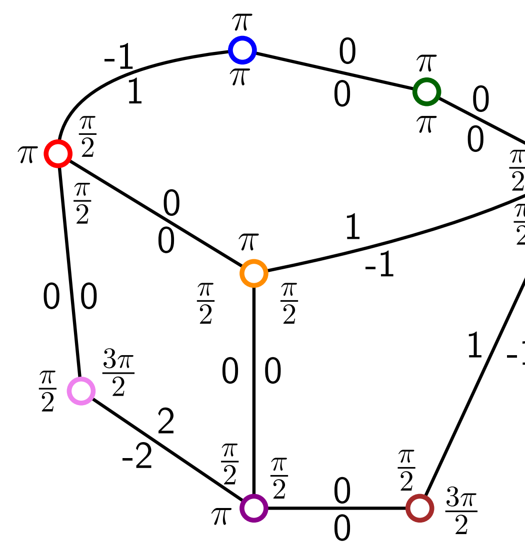


Orthogonal Representations

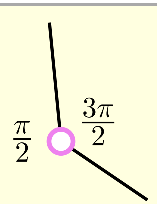
An orthogonal representation of a plane graph G represents an equivalence class of orthogonal drawings with the same "shape" [Tamassia, 1987].



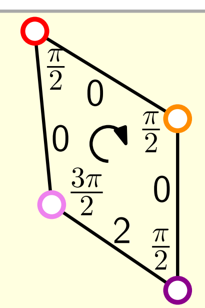
$O(n)$ time



It is a feasible assignment of angles to each vertex-face incidence and of integers to each edge-face incidence, where feasible means:



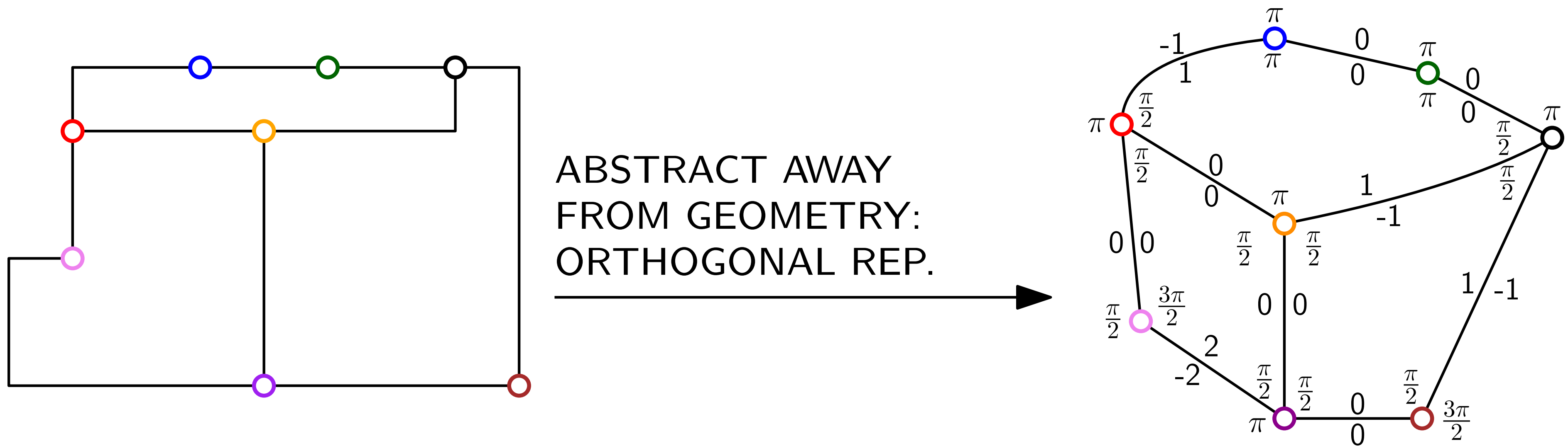
1) The sum of the angles around a vertex is 2π



2) The sum of the angles inside a face comply with an orthogonal polygon $\frac{\pi}{2}+\frac{\pi}{2}+\frac{\pi}{2}+2\cdot\frac{\pi}{2}-\frac{\pi}{2} = \pi(4-2) = 2\pi$

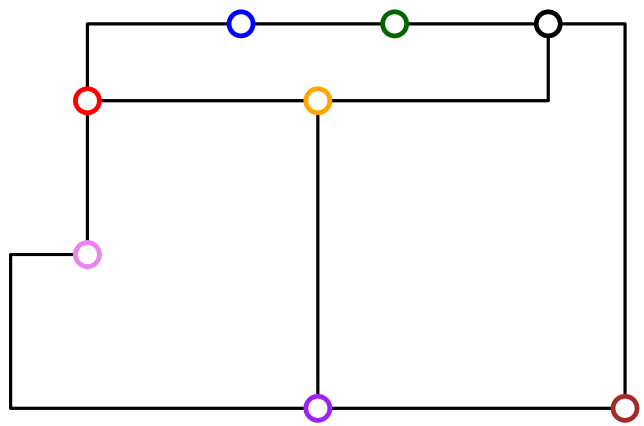
Orthogonal Sketches

GOAL: Define “small” records to be assigned to bags

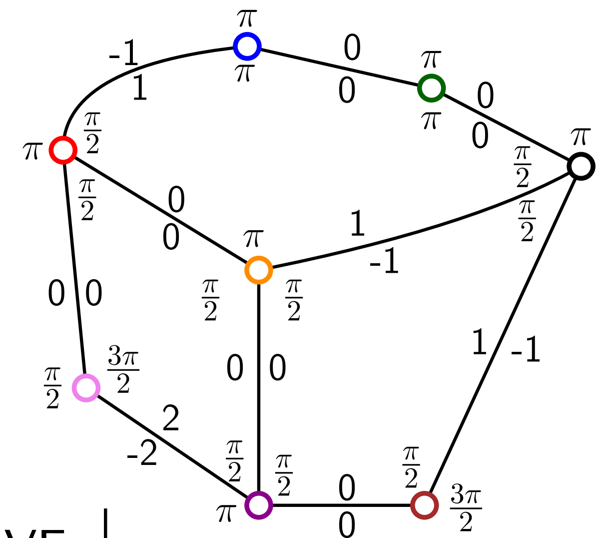


Orthogonal Sketches

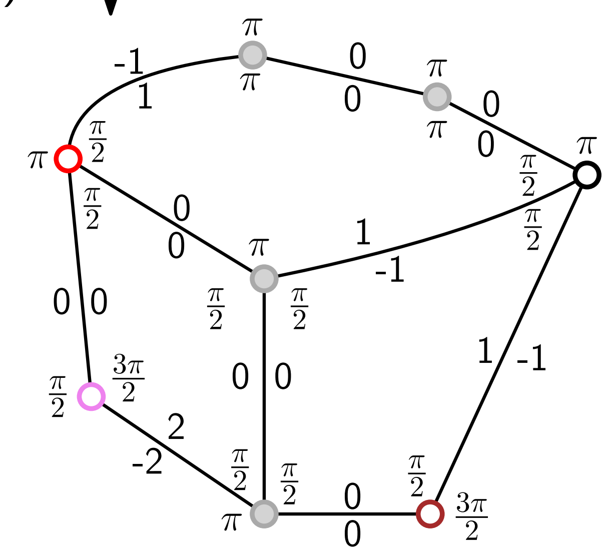
GOAL: Define “small” records to be assigned to bags



ABSTRACT AWAY
FROM GEOMETRY:
ORTHOGONAL REP.

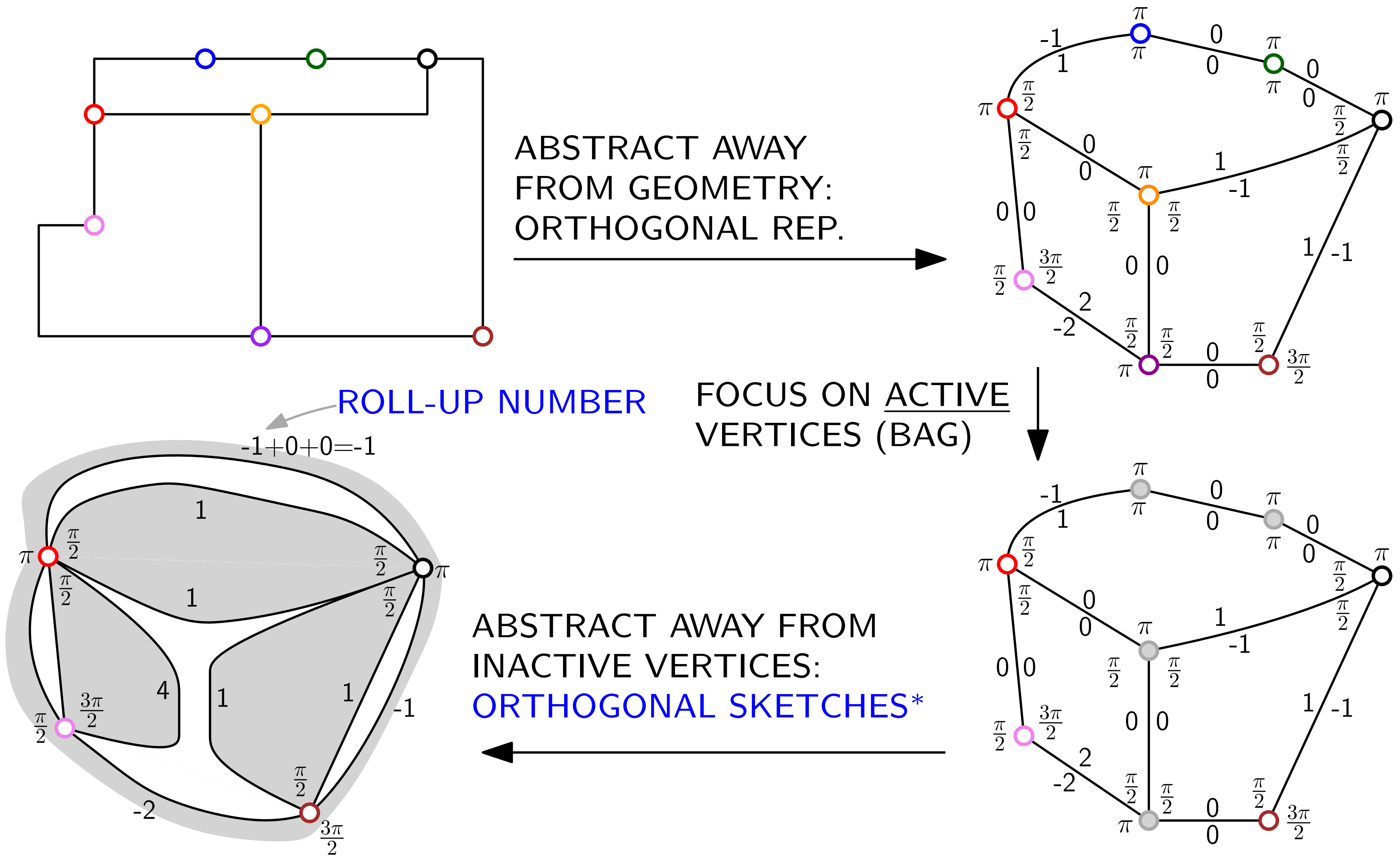


FOCUS ON ACTIVE
VERTICES (BAG)



Orthogonal Sketches

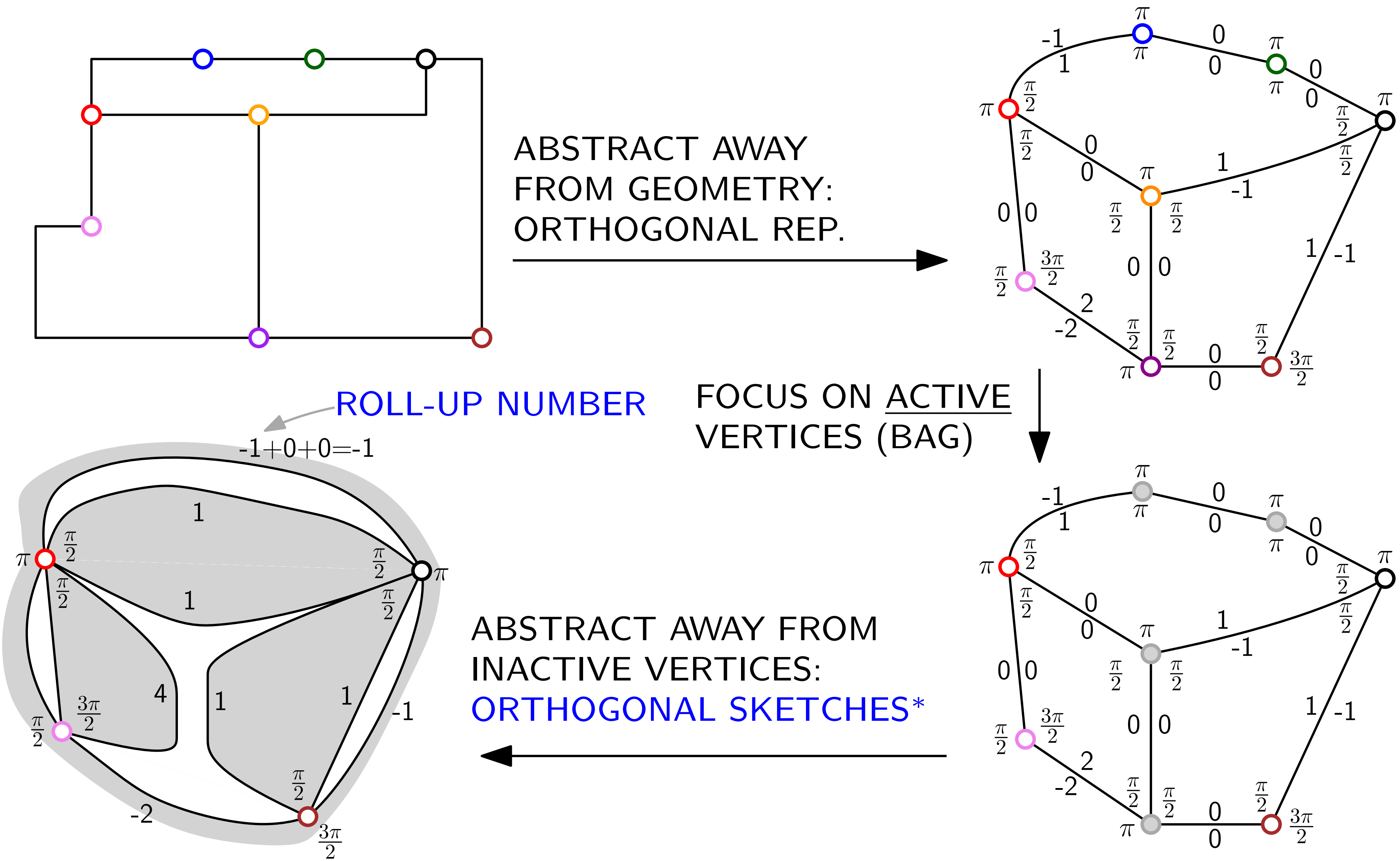
GOAL: Define “small” records to be assigned to bags



* Orthogonal sketches also contain dummy vertices/edges to preserve connectivity

Orthogonal Sketches

Lemma: There are $k^{O(k)}(\sigma + b)^k$ distinct orthogonal sketches.



* Orthogonal sketches also contain dummy vertices/edges to preserve connectivity

Orthogonal Sketches

Lemma: There are $k^{O(k)}(\sigma + b)^k$ distinct orthogonal sketches.

- $k^{O(k)}$ possible embeddings on $O(k)$ vertices

Orthogonal Sketches

Lemma: There are $k^{O(k)}(\sigma + b)^k$ distinct orthogonal sketches.

- $k^{O(k)}$ possible embeddings on $O(k)$ vertices
  - For each embedding we count:
    * Possible vertex-face angle assignments are $2^{O(k)}$

Orthogonal Sketches

Lemma: There are $k^{O(k)}(\sigma + b)^k$ distinct orthogonal sketches.

- $k^{O(k)}$ possible embeddings on $O(k)$ vertices
  - For each embedding we count:
    * Possible vertex-face angle assignments are $2^{O(k)}$
    * Possible roll-up number assignments are $(\sigma + b)^k$

Orthogonal Sketches

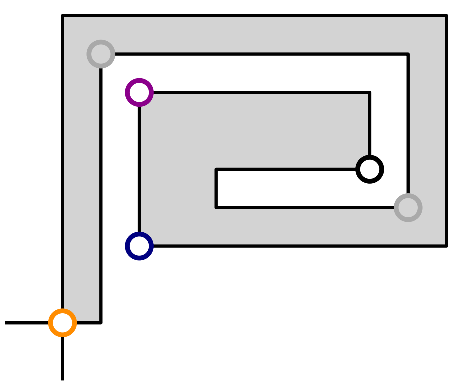
Lemma: There are $k^{O(k)}(\sigma + b)^k$ distinct orthogonal sketches.

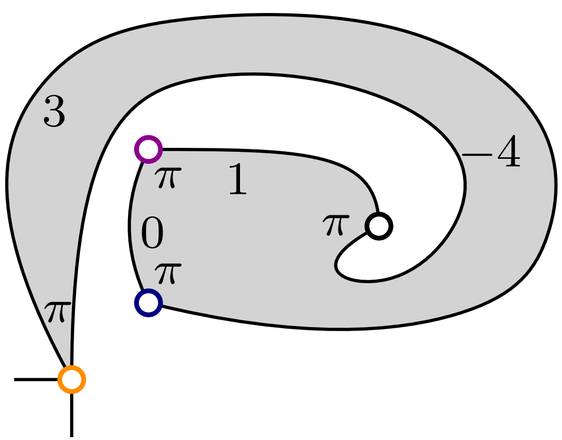
- $k^{O(k)}$ possible embeddings on $O(k)$ vertices
  - For each embedding we count:
    * Possible vertex-face angle assignments are $2^{O(k)}$
    * Possible roll-up number assignments are $(\sigma + b)^k$

$|\rho(u, v)| \leq \sigma + b + 4$

ROLL-UP NUMBER

how much a facial path rolls up depends on the number of bends and degree-2 vertices in the face





Orthogonal Sketches

Lemma: There are $k^{O(k)}(\sigma + b)^k$ distinct orthogonal sketches.

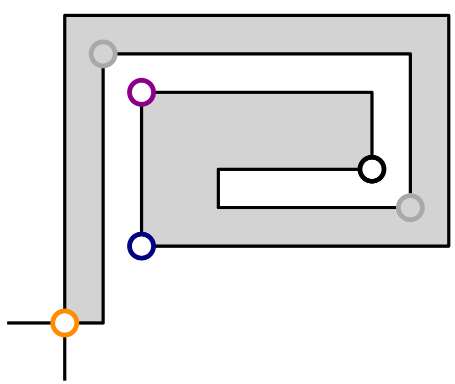
- $k^{O(k)}$ possible embeddings on $O(k)$ vertices
  - For each embedding we count:
    - * Possible vertex-face angle assignments are $2^{O(k)}$
    - * Possible roll-up number assignments are $(\sigma + b)^k$

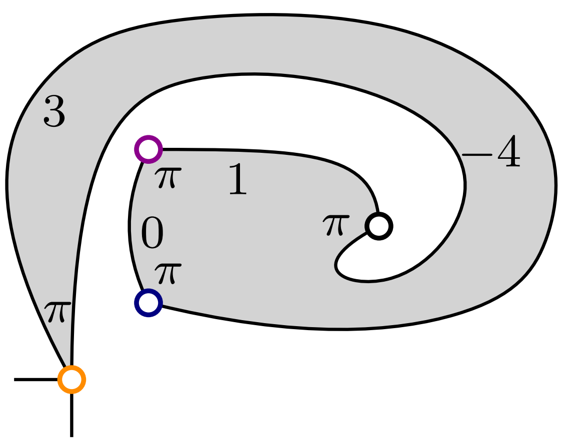
ROLL-UP NUMBER

$|\rho(u, v)| \leq \sigma + b + 4$

how much a facial path rolls up depends on the number of bends and degree-2 vertices in the face

Roll-up numbers are not independent of each another, choosing k of them along a spanning tree fixes all the others





Introduce Operation

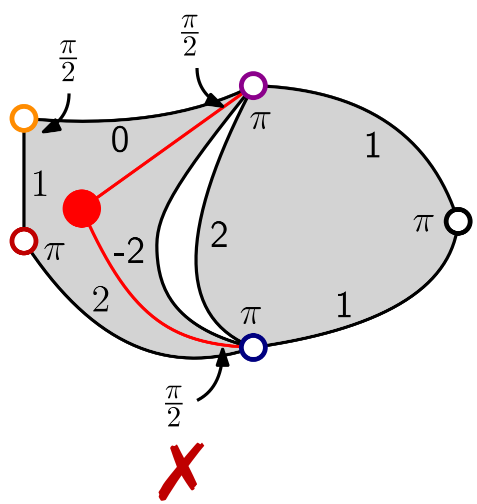
INPUT: a set S of orthogonal sketches of a bag X and a vertex ●

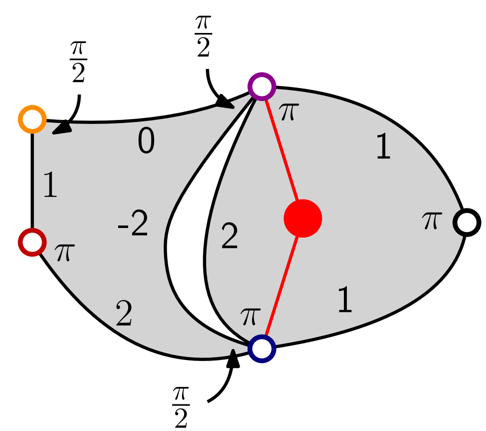
Introduce Operation

INPUT: a set S of orthogonal sketches of a bag X and a vertex ●

for each orthogonal sketch $o \in S$:

identify faces where ● can be placed;

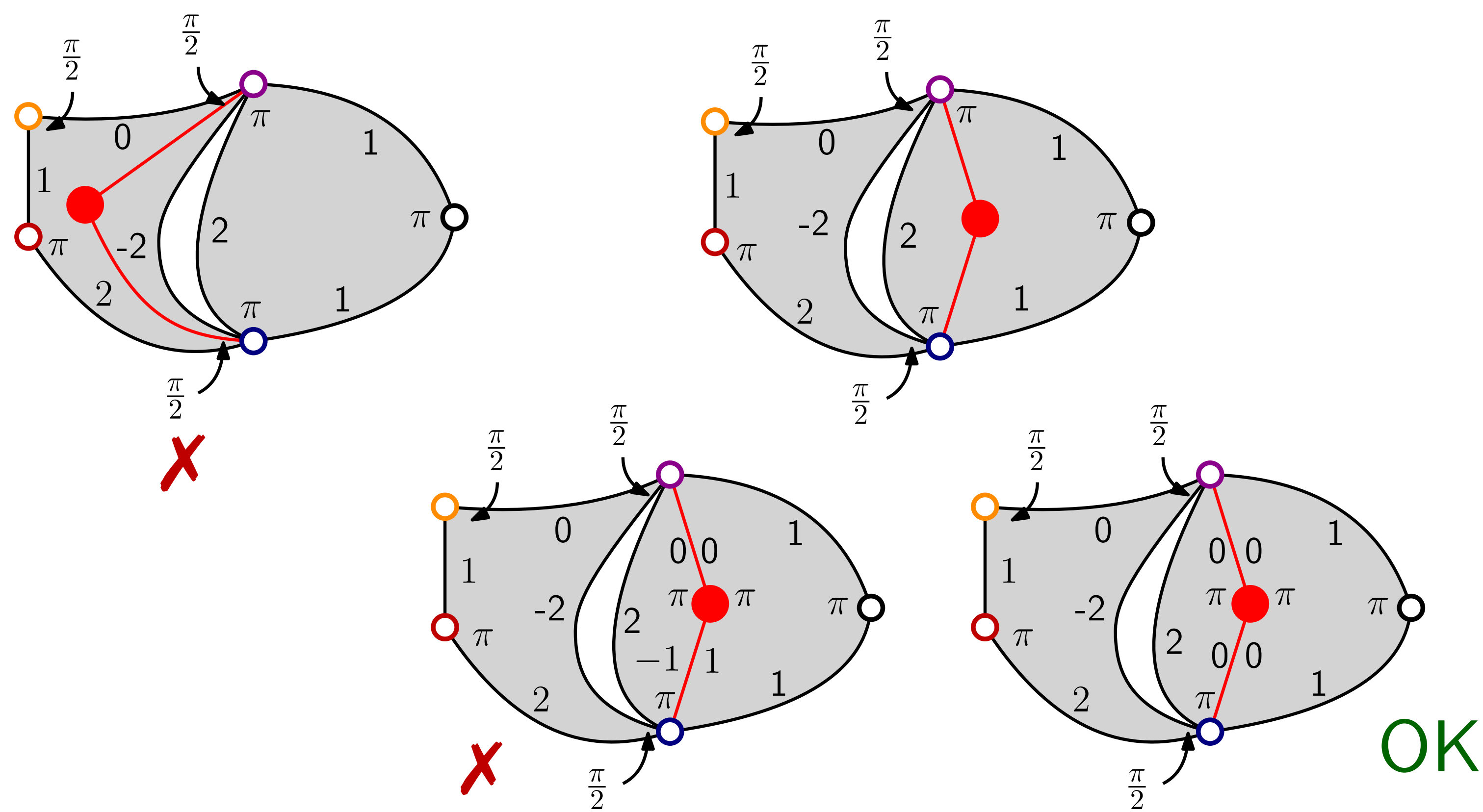




Introduce Operation

INPUT: a set S of orthogonal sketches of a bag X and a vertex ●

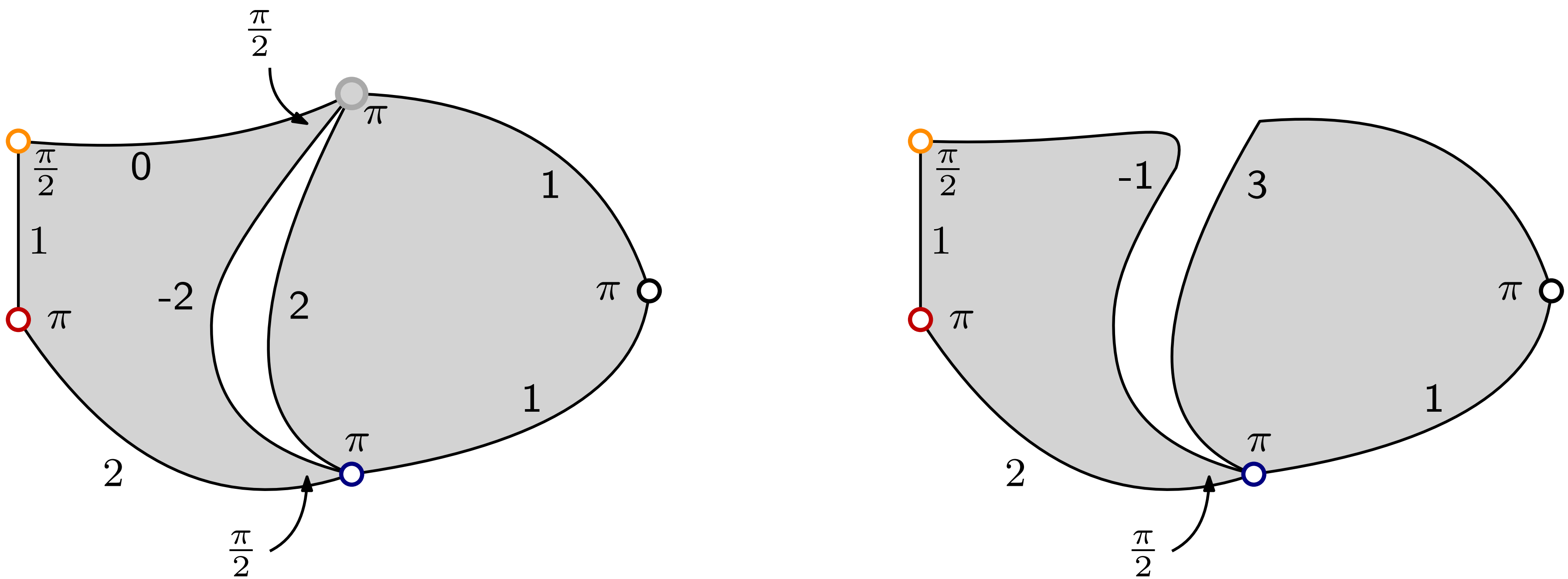
- for each orthogonal sketch $o \in S$:
  - identify faces where ● can be placed;
  - foreach planar embedding:
    - generate all roll-up number assignments for the new edges;
    - keep only the shapes that are valid;



Forget Operation

INPUT: a set S of orthogonal sketches of a bag X and a vertex ●

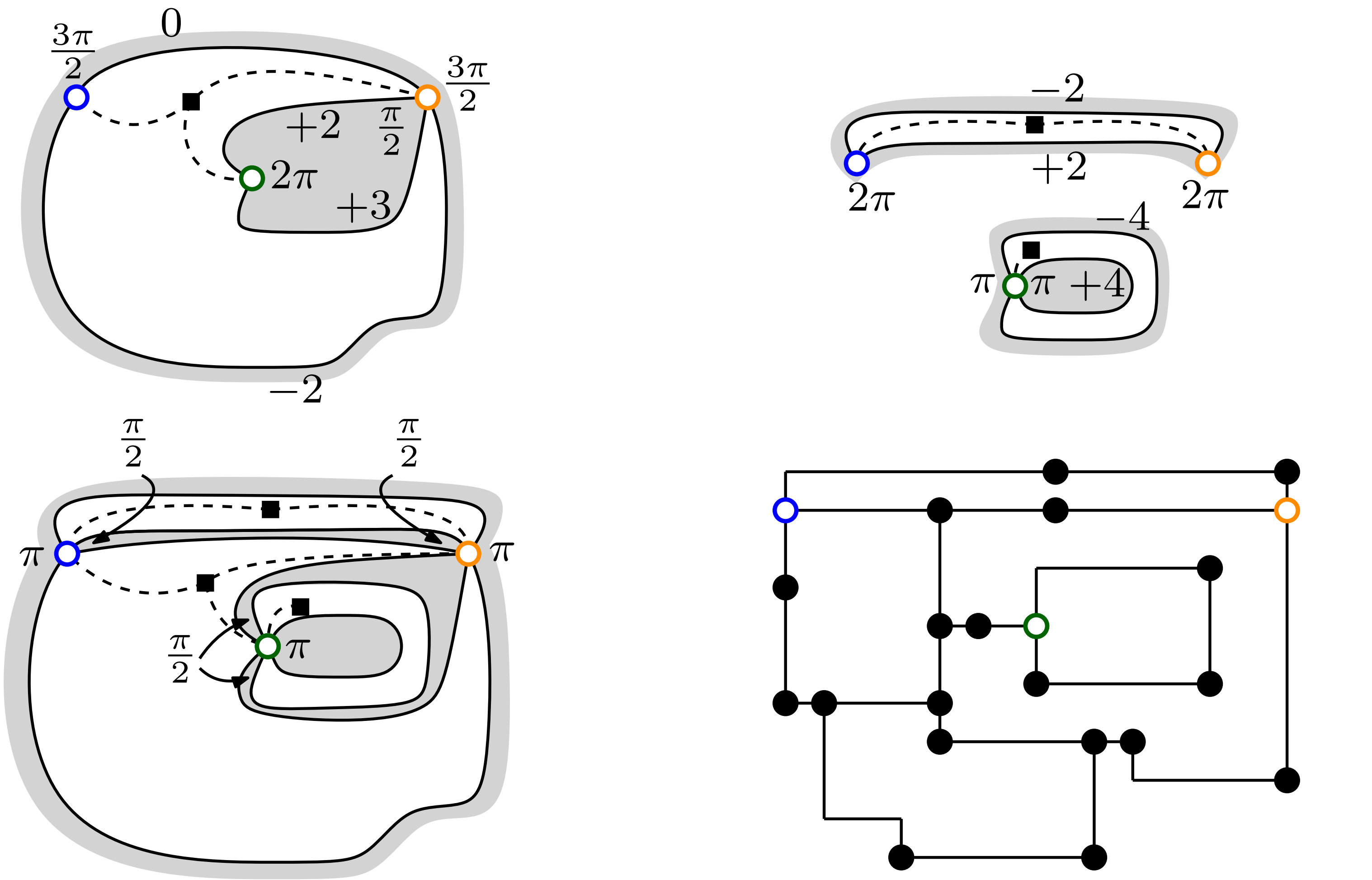
Update each orthogonal sketch $o \in S$



Join Operation

INPUT: 2 sets of orthogonal sketches to be merged at a bag X

More complex procedure to ensure efficiency.



Open Problems & Future Work

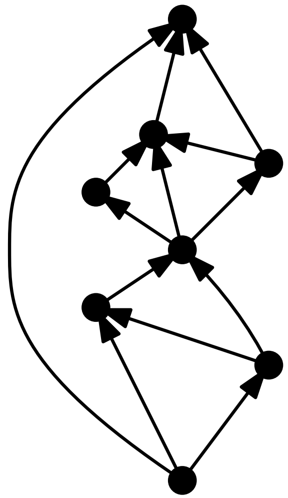
Open Problems & Future Work

1. FPT algorithm for OrthogonalPlanarity parametrized by treewidth and number of bends?

2. Subcubic time complexity for series-parallel graphs?

3. UpwardPlanarity and WindrosePlanarity admit similar combinatorial characterizations based on vertex-angles.

   Our approach can be extended to prove that these problems can be solved in polynomial time on graphs of bounded treewidth.



THANK YOU!